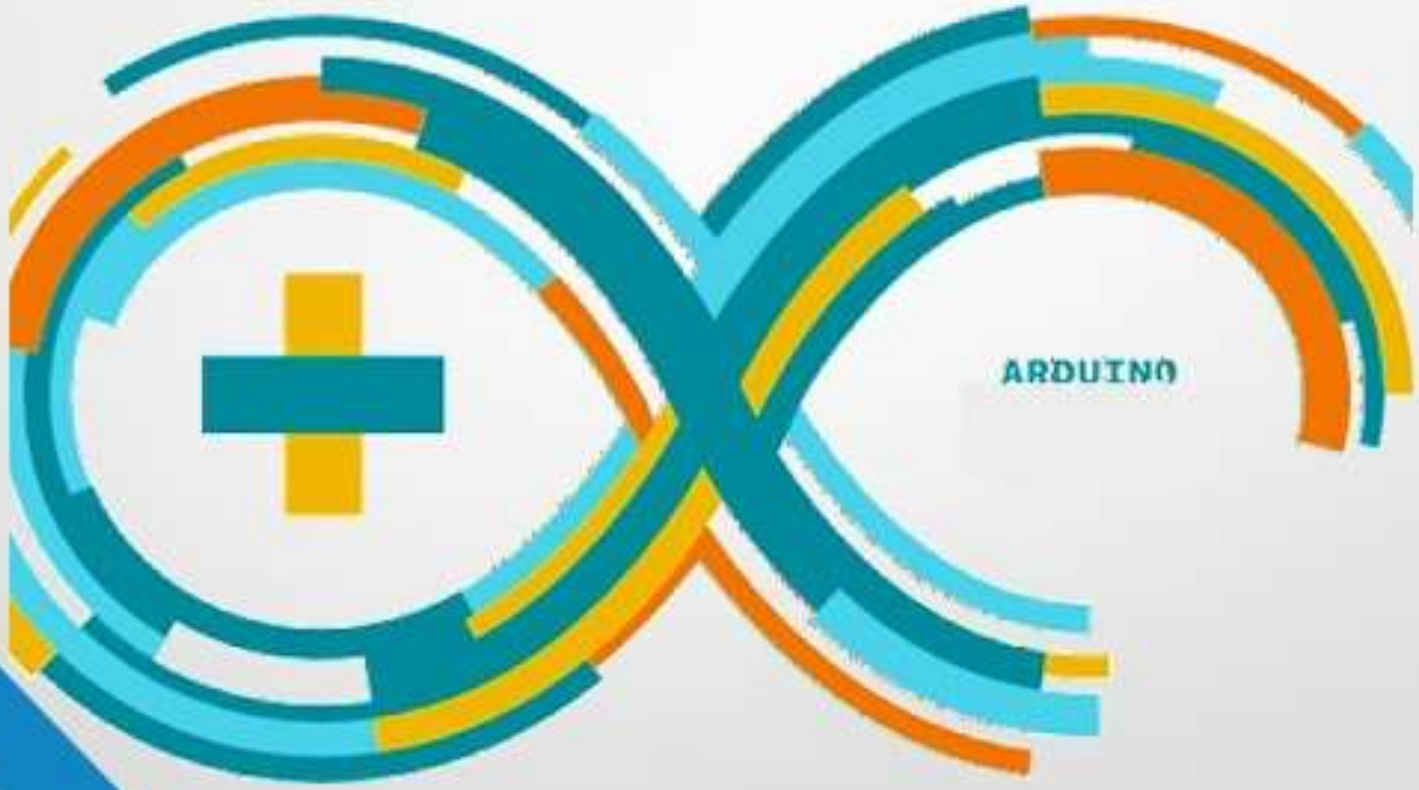
EMBEDDED SYSTEM (Arduino microcontroller)



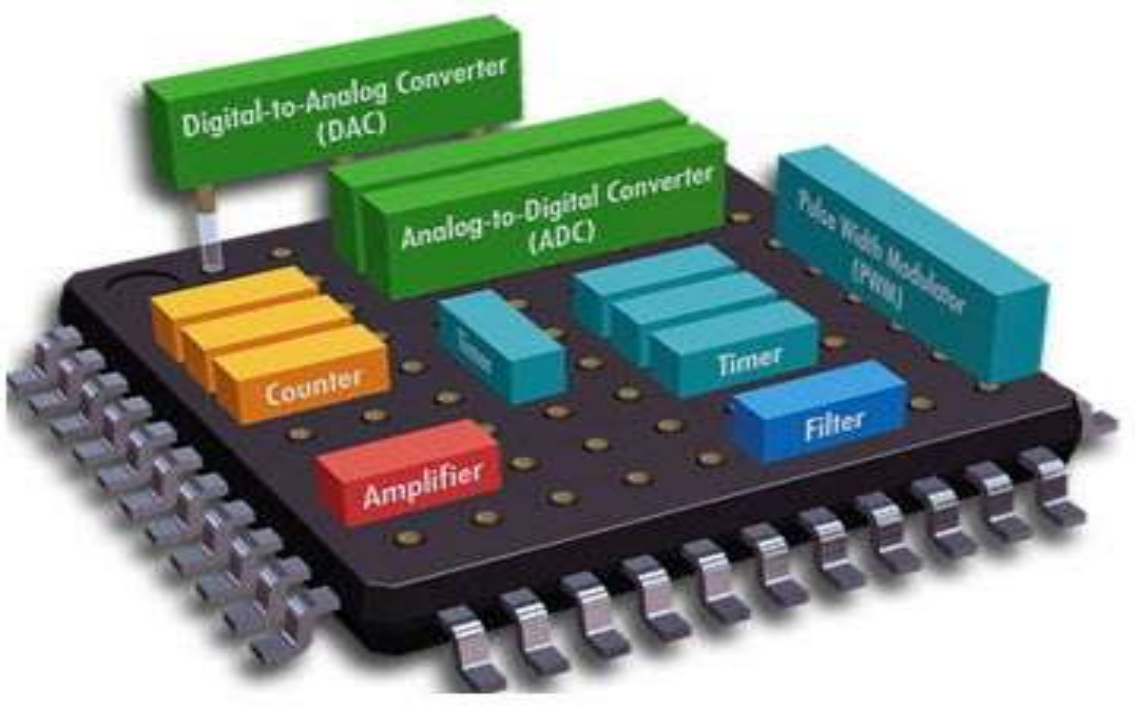
Contents

Introduction

- An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints.
- Arduino is an open-source electronic prototyping platform based on a simple i/o board and a development environment for writing software for the board.
- It is an open source hardware, any one can get the details of its design and modify it or make his own one himself.

Embedded system

- An embedded system is a combination of computer hardware and software, either fixed in capability or programmable



Overview

What is Arduino?

- What is it used for?
- How to get started
- Demonstration

- Questions are welcome at any time.

Arduino is a platform

- A physical Input / Output board (I/O) with a programmable Integrated Circuit (IC).

What is it used for?

- Physical Computing projects / research
- Interactive Installations
- Rapid prototyping

- When you wish to move beyond the traditional Mouse, Keyboard and Monitor to develop novel and custom interactions in your project work.

What can it do?

- Sensors (to sense stuff)
  - Push buttons, touch pads, tilt switches.
  - Variable resistors (eg. volume knob / sliders)
  - Photoresistors (sensing light levels)
  - Thermistors (temperature)
  - Ultrasound (proximity range finder)
- Actuators (to do stuff)
  - Lights, LED's
  - Motors
  - Speakers
  - Displays (LCD)

Why Arduino?

- It is Open Source, both in terms of Hardware and Software.
- It is cheap(1300रु), the hardware can be built from components or a prefab board can be purchased for approx 900रु.
- It can communicate with a computer via serial connection over USB.
- It can be powered from USB or standalone DC power.

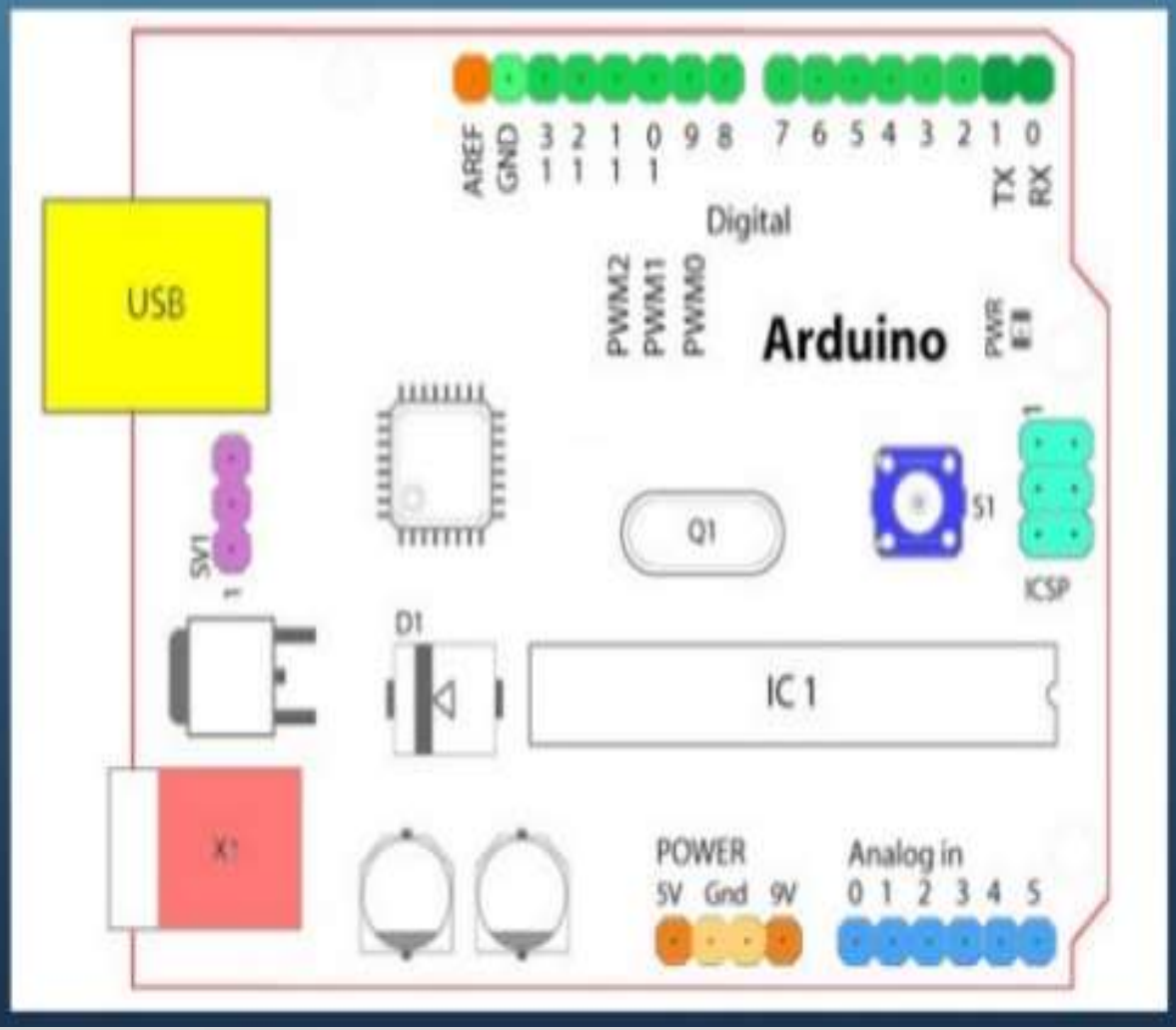
Why Arduino?

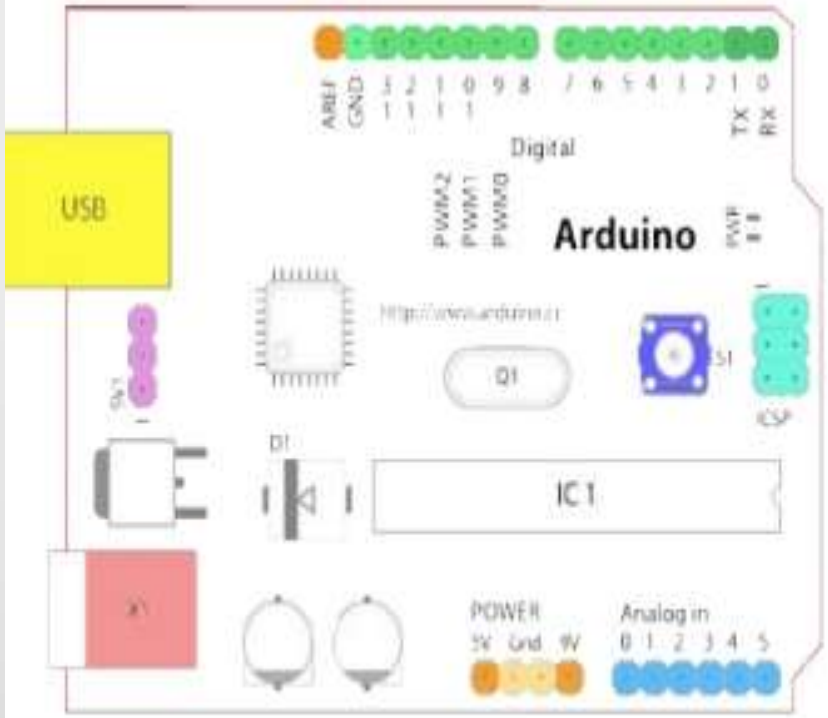
- It can run standalone from a computer (chip is programmable) and it has memory (a small amount).
- It can work with both Digital and Analog electronic signals. Sensors and Actuators.
- You can make cool stuff! Some people are even making simple robots, and we all know robots are just cool. ☺

How to get started

- You'll need a board of course, along with the USB cable and DC power supplies.
- Read about, understand what you are working with and download the IDE: http://www.arduino.cc
- Mac, Windows and Penguin friendly versions available
- Then you are ready to plug it in!

Architecture





Layout of an Arduino

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX (dark green)

  These pins cannot be used for digital i/o (digitalRead and digitalWrite) if you are also using serial communiation (e.g. Serial.begin).
- Reset Button - S1 (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

- IC1 – Microcontroller

ICSP – In circuit serial programming

AREF – Analog reference pin

GND – Digital ground

2-13 Digital pins

0-1(TX/RX) – Serial In/Out

S1 – Reset button

0-5 Analog Input pins

Power and Ground pins

Arduino Family

Arduino Uno

Arduino Leonardo

Arduino Mega ADK

Arduino Ethernet

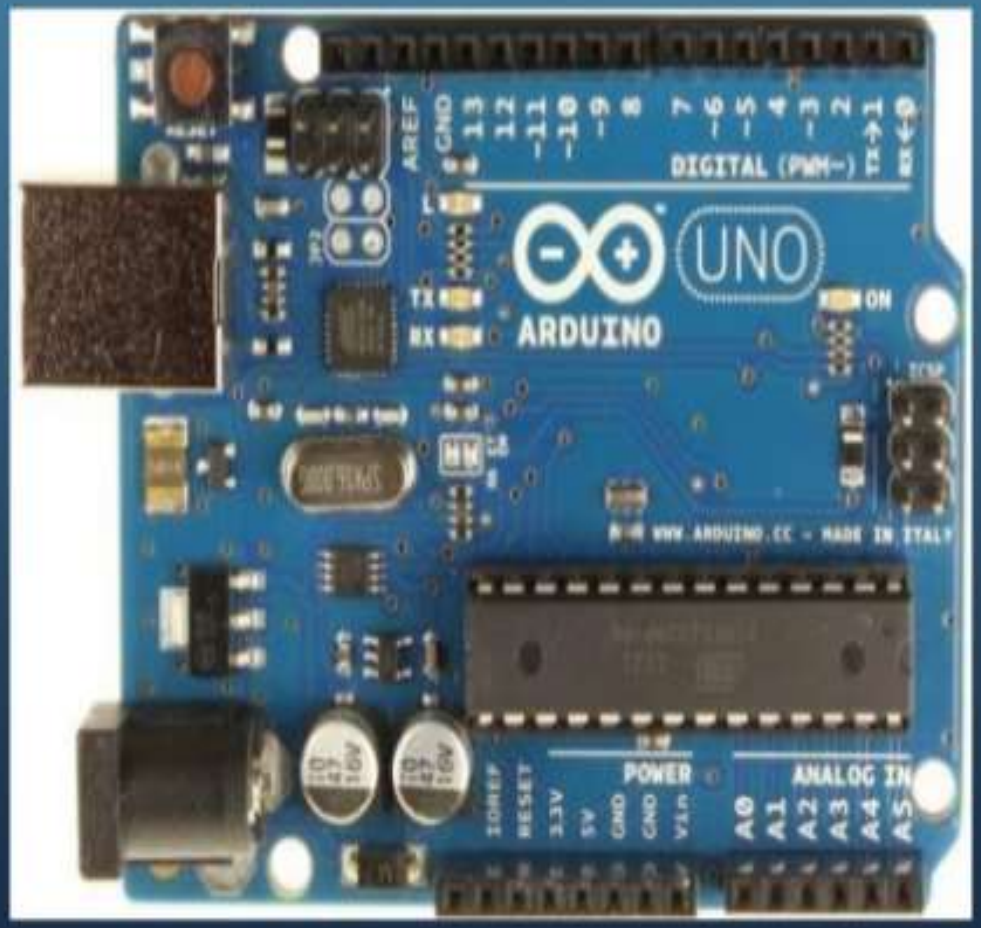
Arduino Due

Arduino Yún

Arduino Mega 2560

Arduino Mini

Arduino example

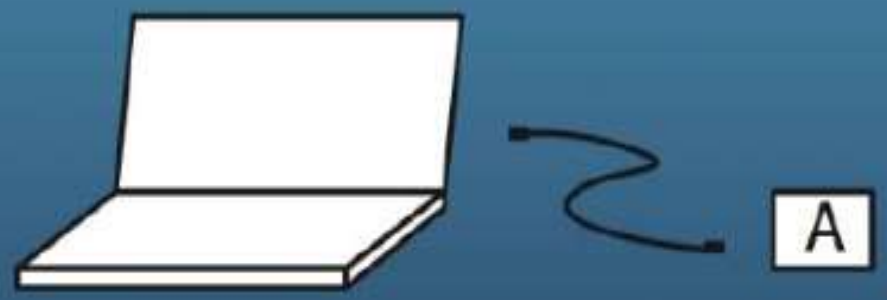


Basic Process

- Design the circuit:
  - What are electrical requirements of the sensors or actuators?
  - Identify inputs (analog inputs)
  - Identify digital outputs
- Write the code
  - Build incrementally
    - Get the simplest piece to work first
    - Add complexity and test at each stage
    - Save and Backup frequently
  - Use variables, not constants
  - Comment liberally

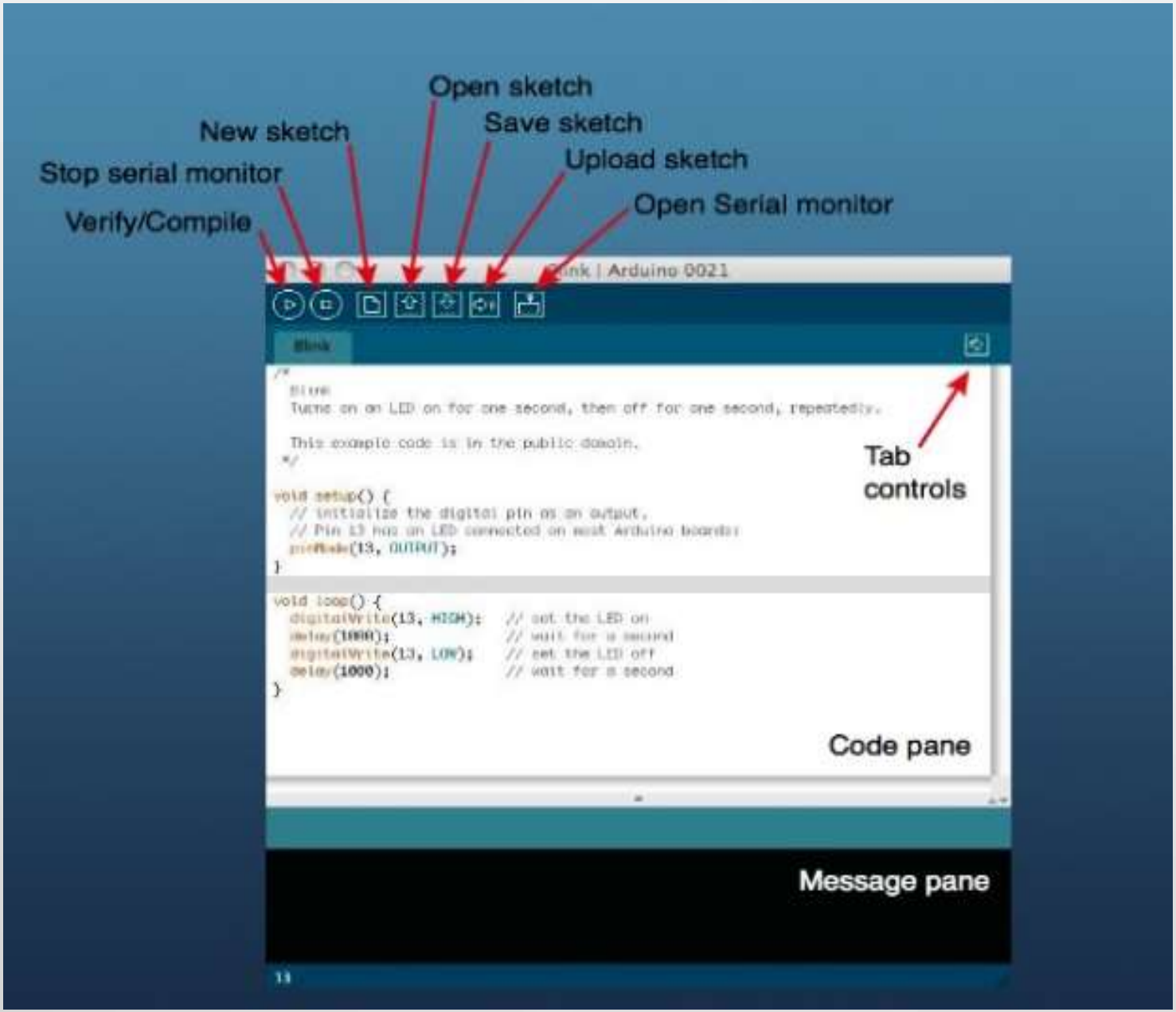
Writing and Downloading Code

Write sketch on PC



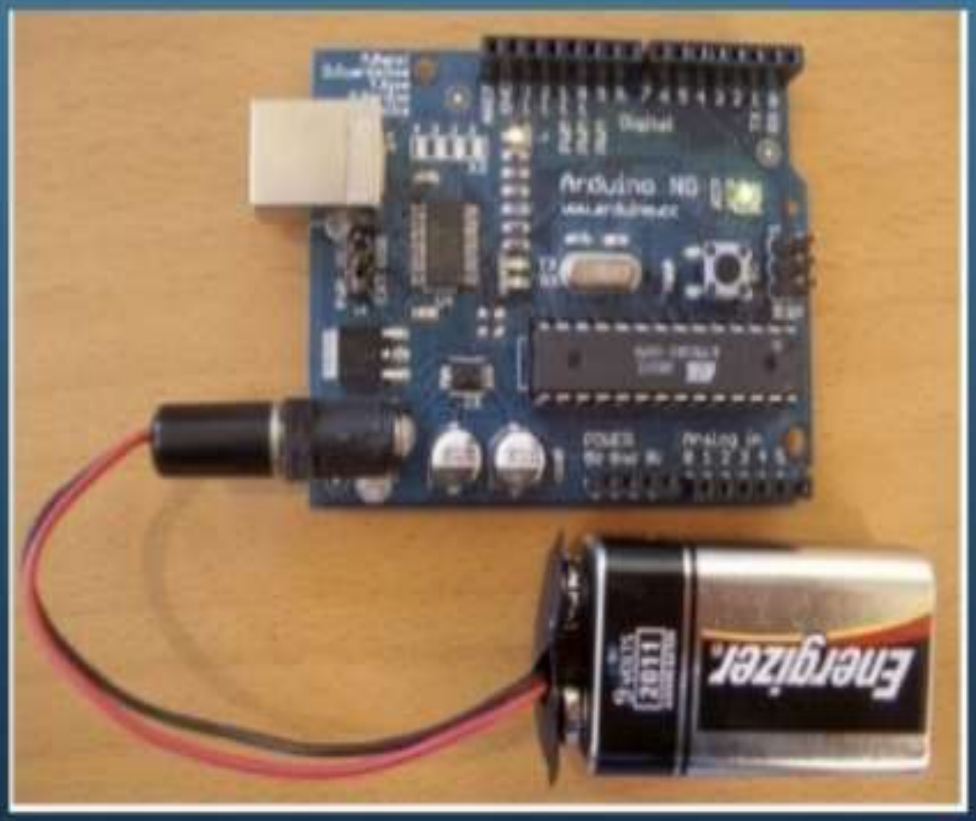
Download sketch to Arduino

•The main headings are “File” “Edit” “Sketch” “Tools” “Help” and several shortcut icons beneath “Verify”, “Upload”, “New”, “Open”, “Save”, and at the far right, the “Serial Monitor”.

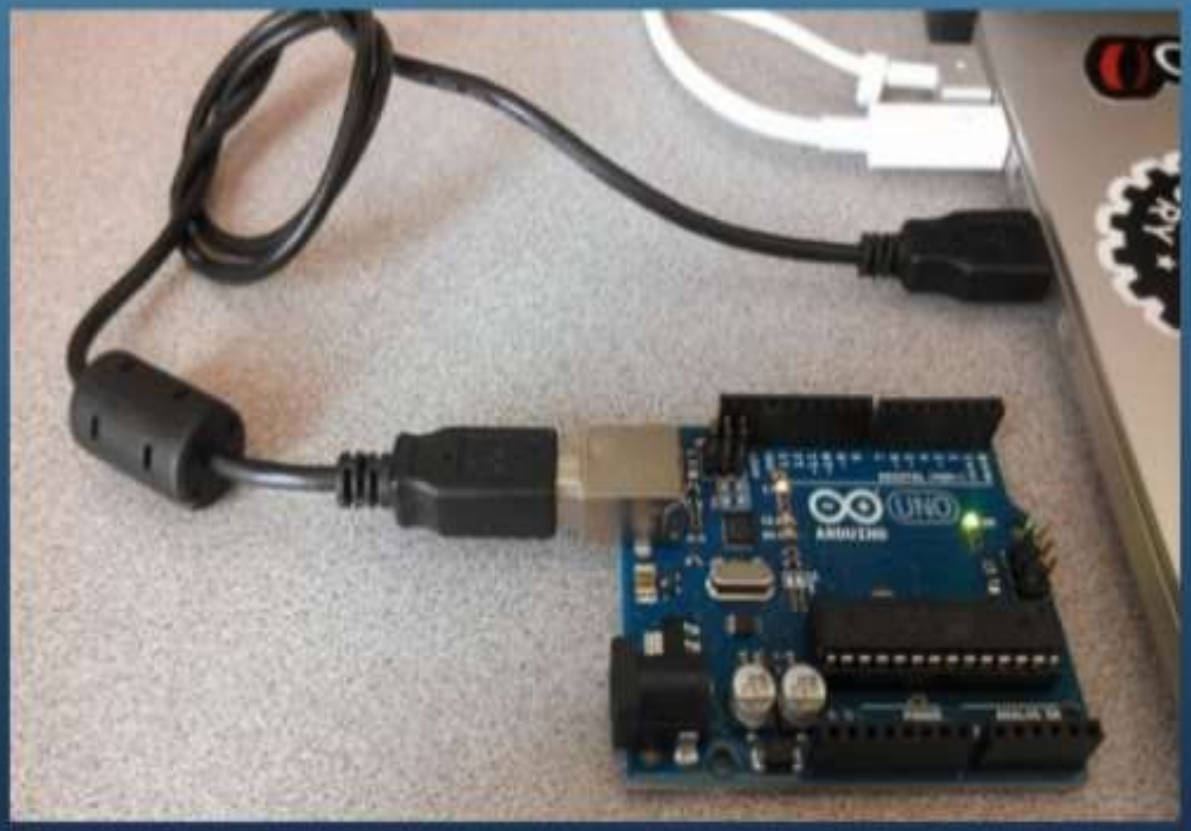

Open sketch
New sketch
Save sketch
Stop serial monitor
Upload sketch
Open Serial monitor
Verify/Compile
Blink | Arduino 0021
Blink
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
This example code is in the public domain.
*/
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected on most Arduino boards:
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH); // set the LED on
delay(1000); // wait for a second
digitalWrite(13, LOW); // set the LED off
delay(1000); // wait for a second
}
Tab controls
Code pane
Message pane


sketch_aug09a | Arduino 1.0.1
File Edit Sketch Tools Help
sketch_aug09a

- X1 – External power supply or a battery.The power requirement for ARDUINO is 9 to 12V DC,250mA or more,2.1mm plug

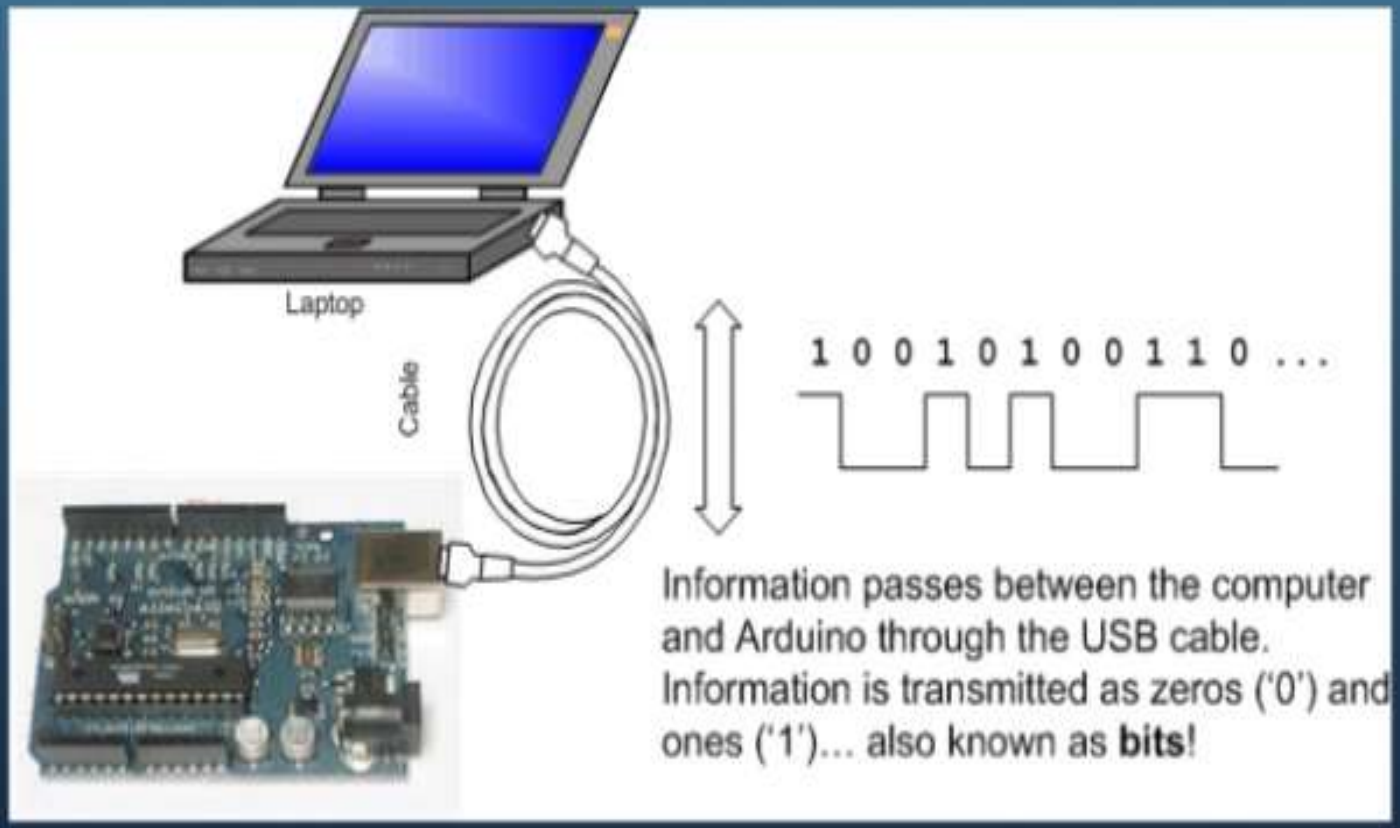


- USB – We can connect arduino to our PC for programmming the arduino through USB.



Serial communication

- The communication between the computer and the arduino is 'serial' , because data is broken down into bits, each sent one after the other down a single wire.



Programming the arduino

- Arduino programs only need to define two functions to make a runnable program:

  void setup() {} – a function run once at the start of a program that can initialize settings. This section is widely used to initialize variables, pin modes, set the serial baud rate and related.

  void loop() {} – a function called repeatedly until the board powers off. This section is the part of the code that loops back onto itself and is the main part of the code.
- Programmers are free to add subroutines using the same syntax:

  void subroutinename() {}
- We will use same data types, operators, statements as we use in C programming language.

Programming

```
void setup()
{
    // put your setup code here, to run once:

}


void loop()
{
    // put your main code here, to run repeatedly:

}
```

Bare minimum code

- setup : It is called only when the Arduino is powered on or reset. It is used to initialize variables and pin modes

- loop : The loop functions runs continuously till the device is powered off. The main logic of the code goes here. Similar to while (1) for micro-controller programming.

PinMode

- A pin on arduino can be set as input or output by using pinMode function.
- pinMode(13, OUTPUT); // sets pin 13 as output pin
- pinMode(13, INPUT); //sets pin 13 as input pin

Reading/writing digital values

- digitalWrite(13, LOW); // Makes the output voltage on pin 13, 0V
- digitalWrite(13, HIGH); // Makes the output voltage on pin 13, 5V
- int buttonState = digitalRead(2); // reads the value of pin 2 in buttonState

What are Libraries?

- Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download.

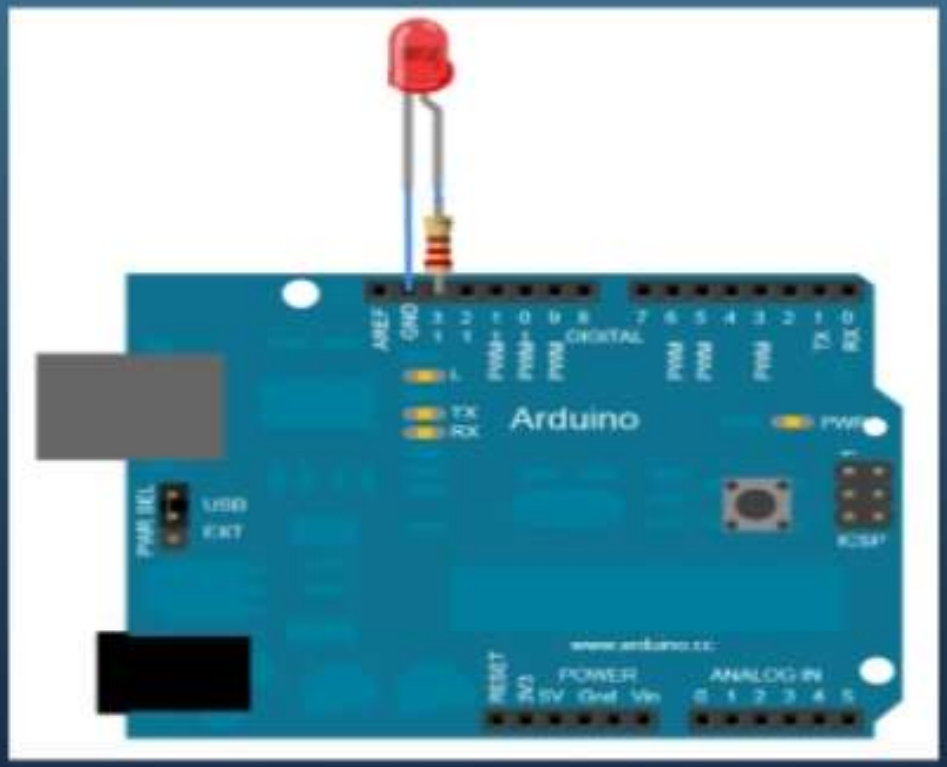
- Almost every line of code needs to end with a semicolon ';'.
- To write single line comments in the code, type two back slashes followed by the text.
- To write multi-line comments, start the comment with /* and end with */.
- The Arduino language is case sensitive.
- The following code represents the minimum in order for a program to compile:

Programming examples

- To output a value on arduino window

❖Blink LED programm

•In this program the Led should blink (turn on) for 1 second and after 1 second the LED should turn off for 1 second,and hence this cycle repeats.



- pinMode(13, OUTPUT);
- digitalWrite(13, HIGH);
- digitalWrite(13, LOW);
- delay(1000);

Blink | Arduino 0021

Blink §

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

set up() function is executed only once at the start

Blink | Arduino 0021

Blink §

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Loop function repeats indefinitely

Applications

- Light control
- Motor control
- Home Automation
- Robotics
- Networking
- Scientific equipment
- Arduinome
- ArduinoPhone
- Water quality testing platform

Conclusion

- Over the years, Arduino has went out to become a huge success . By using the arduino we can put together both software and hardware. Arduino will be the most useful interface between the software and the hardware in future. In summary, this arduino concept is a good software hardware co-design practice.

Thank you

☺

References

- http://www.arduino.cc-Arduino Official webpage
- http://en.wikipedia.org/wiki/Arduino-wikipedia
- http://www.arduino.cc/playground/Projects/ArduinoUsers
- http://www.arduinothedocumentary.org
- http://www.arduinothedocumentary.org
- http://arduino.cc/en/Tutorial/WebServer
- http://slideshare.com/Arduino section programming slides
- "Programming Arduino Getting Started with Sketches" McGraw-Hill.
- Embedded Microcontroller Systems: real time interfacing Jonathan Valvano, 2006
- Beginning arduino by Michael McRobert